Clipper

เภา สิงห์บุปผา

SOFT MAIL

การอินเตอร์เฟสเม้าส์เข้ากับกราฟิกในคลิปเปอร์

แนะนำการอินเตอร์เฟสเม้าส์เข้ากับกราฟิกในคลิปเปอร์ การทำบัททอน และการประยุกต์ไปใช้ในการนำเสนอ

ในตอนที่แล้วผู้เขียนได้แนะนำการรับข้อมูลในโหมดกราฟิกของคลิปเปอร์โดยการนำเอาฟังก์ชันของ DGE มาทำเป็น USER DEFINE FUNCTION เพื่อรับข้อมูลทั้งภาษาอังกฤษระดับเดียวและการรับภาษาไทยซึ่งแสดงการจัดระดับด้วยเพื่อช่วยในการรับข้อมูลภาษาไทย ในตอนนี้ผู้เขียนขอแนะนำการอินเตอร์เฟสเม้าส์มาใช้กับโหมดกราฟิกของคลิปเปอร์ เนื่องจากในปัจจุบันเครื่องคอมพิวเตอร์โดยมากจะมีอุปกรณ์เม้าส์ประกอบอยู่ด้วย หากเราสามารถนำเม้าส์มาใช้ประกอบกับโปรแกรมย่อมเป็นการอำนวยความสะดวกมากขึ้น อีกทั้งในตอนท้ายของบทความนี้จะเป็นการแนะแนวทางในการประยุกต์ใช้คลิปเปอร์ในการนำเสนอเกี่ยวกับฐานข้อมูลรูปภาพ ซึ่งผู้อ่านสามารถนำไปประยุกต์ใช้กับงานในด้านอื่น ๆ ได้

การตรวจสอบเม้าส์ ปุ่มเม้าส์ ขอบเขตการแสดงผล

เพื่อให้เข้าใจง่ายและสามารถนำไปประยุกต์ใช้ได้อย่างรวดเร็ว ผู้เขียนขอให้ผู้อ่านดูโปรแกรมในรูปที่ 1 ซึ่งผลของการรันโปรแกรมจะได้ดังรูปที่ 2 ซึ่งผู้เขียนขออธิบายการทำงานของโปรแกรมและฟังก์ชันต่าง ๆ ประกอบไปด้วยดังนี้

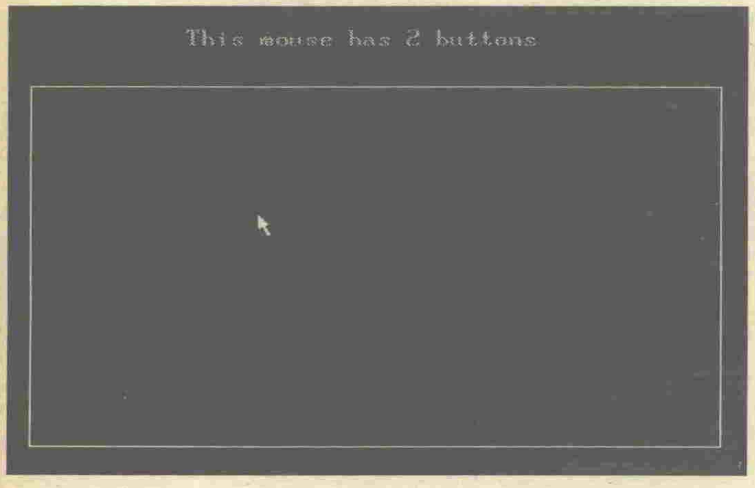
รูปที่ 1 โปรแกรม CHKMOUSE.PRG

```
01 : // Program     : chkmouse.prg
02 : // Prigrammer : Bhao Singboobpha 7/11/93
03 : // Compile     : CLIPPER chkmouse
04 : // Link        : RTLINK FILE chkmouse
05 : // Run         : chkmouse
```

→

รูปที่ 1 (ต่อ)

```
06 : sethires(0)
07 : r  := mreset()
08 : IF r = 0
09 :    saystring(300, 800, 0, 0, 7, "Error - No Mouse Detected")
10 :    INKEY(5)
11 : ELSE
12 :    saystring(300, 650, 0, 0, 7,;
13 :    "This mouse has " + LEFT(LTRIM(STR(r)),1)+" buttons")
14 :    msetwin(100, 100, 998, 600)
15 :    boxfill(100, 100,  900, 500, 20, 14)
16 :    mfixpos(500, 500)
17 :    mcuron()
18 :    DO WHILE .t.
19 :       NButtons := mstatus()
20 :       IF NButtons = 1  .OR. NButtons = 2 .OR. NButtons = 3
21 :          EXIT
22 :       ENDIF
23 :    ENDDO
24 :    mcuroff()
25 : ENDIF
26 : settext()
27 : RETURN
```



รูปที่ 2 ผลที่ได้จากการรันโปรแกรม CHKMOUSE.PRG

รายละเอียดของโปรแกรมมีดังนี้

1. บรรทัดที่ 6 เป็นการเข้าสู่โหมดกราฟิก

2. บรรทัดที่ 7 เป็นการใช้ฟังก์ชัน mreset( ) ซึ่งจะรีเทอร์นค่า 0 หากไม่พบว่ามีการติดตั้งเม้าส์ หากพบว่ามีการติดตั้งเม้าส์จะให้ค่าตัวเลขที่เป็นจำนวนปุ่มกดของเม้าส์

3. บรรทัดที่ 8-10 เมื่อไม่ได้ติดตั้งเม้าส์โปรแกรมจะแจ้งให้ทราบเป็นเวลา 5 วินาที และจะออกจากโปรแกรม

4. บรรทัดที่ 12-17 โปรแกรมจะแจ้งจำนวนปุ่มเม้าส์ กำหนดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์โดยฟังก์ชัน msetwin บรรทัดที่ 15 เป็นการกำหนดกรอบสี่เหลี่ยมที่เม้าส์จะเคลื่อนที่ไปได้ (ดูรูปที่ 3) บรรทัดที่ 16 เป็นการกำหนดตำแหน่งเริ่มต้นของเม้าส์โดยฟังก์ชัน mfixpos(x, y) บรรทัดที่ 17 เป็นการกำหนดให้แสดงเคอร์เซอร์เม้าส์

5. บรรทัดที่ 18-23 มีฟังก์ชัน mstatus( ) เป็นตัวเช็คการกดปุ่มเมาส์ หากมีการกดปุ่มเมาส์ก็จะหลุดจากลูปออกจากโปรแกรมสู่สภาวะเท็กซ์โหมด ในตอนนี้หากเราเลื่อนเม้าส์จะทำให้เคอร์เซอร์เม้าส์เปลี่ยนตำแหน่ง แต่จะถูกจำกัดขอบเขตด้วยฟังก์ชัน msetwin( ) (ดังในรูปที่ 4)

สำหรับจุดประสงค์ของโปรแกรมที่ 1 คือ การเช็คการติดตั้งเม้าส์ การกำหนดขอบเขตการเคลื่อนที่ของเม้าส์ และการเช็คการกดปุ่มเม้าส์

การตรวจสอบตำแหน่งเม้าส์และปุ่มกด

จากบทความตอนที่ผ่านมาจะเห็น

```
msetwin( X0, Y0, X1, Y1 )
     X0          Bottom left X
     Y0          Bottom left Y
     X1          Upper right X
     Y1          Upper right Y
Return value:
       Zero
```

รูปที่ 3 ฟังก์ชัน msetwin จะกำหนดขอบเขตการเคลื่อนที่ของเคอร์เซอร์

```
mstatus ()
   Return value:
       Mouse button status
          0          no button pressed
          1          left button
          2          right button
          3          both buttons
```

รูปที่ 4 ฟังก์ชัน msetwin( ) จะจำกัดขอบเขตการเปลี่ยนตำแหน่งเคอร์เซอร์เม้าส์

ว่า เมื่อเข้าสู่สภาวะกราฟิกโหมดนั้น แนวนอนของจอภาพ (แกน X) จะมีค่าจากซ้ายไปขวาตั้งแต่ 0-1350 ส่วนในแนวตั้งของจอภาพ (แกน Y) จะมีค่าจากล่างไปบนตั้งแต่ 0-1000 เมื่อเรานำความสัมพันธ์ของการกดปุ่มเม้าส์ซึ่งจะมีอยู่ 3 สภาวะคือ กดปุ่มซ้าย กดปุ่มขวาหรือการกดพร้อมกันทั้งสองปุ่มมา และตำแหน่งของการกดว่าอยู่ที่ตำแหน่งใดของจอภาพมาเป็นตัวเลือกกำหนดการทำงานของโปรแกรมต่อไป โปรแกรมในรูปที่ 5 จะเป็นตัวอย่างโปรแกรมที่แสดงตำแหน่งของเคอร์เซอร์เม้าส์และแสดงการกดปุ่มซ้ายหรือปุ่มขวาของเม้าส์ ในการออกจากโปรแกรมจะต้องกดเม้าส์ทั้งสองปุ่มพร้อมกัน

รายละเอียดของโปรแกรมมีดังนี้

1. บรรทัดที่ 7 ติดตั้งเม้าส์เข้ากับโปรแกรม บรรทัดที่ 8-9 กำหนดกรอบสี่เหลี่ยมและขอบเขตการเคลื่อนที่ของเม้าส์ บรรทัดที่ 10 กำหนดจุดเริ่มต้นของเคอร์เซอร์เม้าส์ พร้อมทั้งคำอธิบายว่าให้กดปุ่มเมาส์สองปุ่มเพื่อออกจากโปรแกรม

2. บรรทัดที่ 14 และ 29 เป็นการวนลูปเพื่อเช็คค่าที่รีเทอร์นจากการ

รูปที่ 5 โปรแกรม CHKBUTON.PRG แสดงตำแหน่งเม้าส์และการกดปุ่มเม้าส์

```
01 : // Program    : chkbuton.prg
02 : // Programmer : Bhao Singboobpha 7/11/93
03 : // Compile    : CLIPPER chkbuton
04 : // Link       : RTLINK FILE chkbuton
05 : // Run        : chkbuton
06 : sethires(0)
07 : r  := mreset()
08 : msetwin(100, 100, 998, 600)
09 : boxfill(100, 100,  900, 500, 20, 14)
```

→

รูปที่ 5 (ต่อ)

```
10 : mfixpos(500, 500)
11 : saystring(320, 670, 0, 0, 14, "Press both buttons to Quit.")
12 : saystring(320, 630, 0, 0, 14, "Position =  500,  500")
13 : mcuron()
14 : DO WHILE .t.
15 :       NButtons := mstatus()
16 :       IF   NButtons = 3
17 :            EXIT
18 :       ELSEIF NButtons = 1
19 :            saystring(320, 710, 0, 0, 14, "Press left botton. ")
20 :            TONE(330,2)
21 :       ELSEIF NButtons = 2
22 :            saystring(320, 710, 0, 0, 14, "Press right botton.")
23 :            TONE(660,2)
24 :       ELSEIF mmotion() <> 0
25 :       xpos := mgetx()
26 :       ypos := mgety()
27 :       saystring(510, 630, 0, 0, 14, str(xpos,5)+","+str(ypos,5))
28 :    ENDIF
29 : ENDDO
30 : TONE(330,2)
31 : mcuroff()
32 : settext()
33 : RETURN
```

กระทำของเม้าส์ เช่น การเคลื่อนที่ การกดปุ่มเม้าส์

3. ในบรรทัดที่ 15 เป็นการรับค่าจากการกดปุ่มเม้าส์โดยฟังก์ชัน mstatus( ) แล้วส่งค่าไปเช็คที่บรรทัด 16, 18 และ 21 โดยหากเม้าส์ถูกกดทั้งสองปุ่มพร้อมกัน mstatus( ) = 3 และหลุดจากลูปออกจากโปรแกรม (เช็คที่บรรทัด 16 และ 17) หากกดปุ่มซ้าย

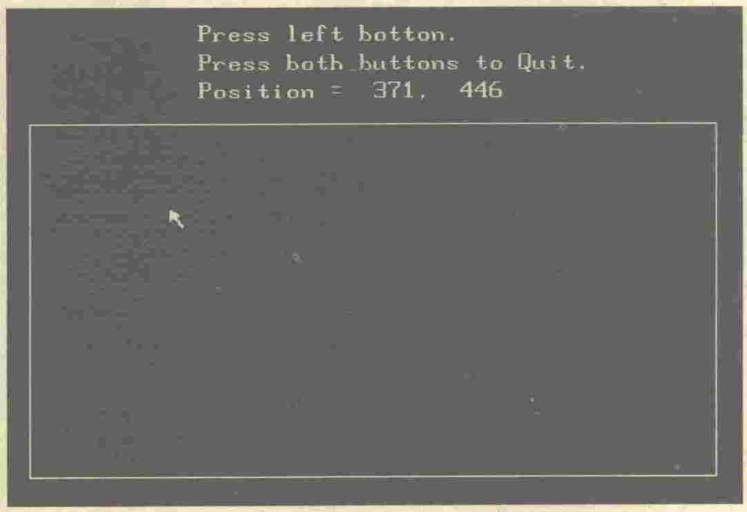
```
mmotion()
Return value:
   0     No motion since last call
   1     The mouse has moved since last call
```

รูปที่ 6 ฟังก์ชัน mmotion( ) จะทำการตรวจสอบเมื่อเม้าส์เคลื่อนไหว

```
mgetx()
Return value:
Mouse cursor's X position in dGE screen units, 0-1350.

mgety()
Return value:
Mouse cursor's Y position in dGE screen units, 0-1000.
```

รูปที่ 7 ฟังก์ชัน mgetx( ) และ mgety( ) จะตรวจสอบตำแหน่งเม้าส์ในแนวแกน x และ y เมื่อเม้าส์เคลื่อนที่



รูปที่ 8 ผลจากการรันโปรแกรมจากรูปที่ 5

mstatus( ) = 1 โปรแกรมจะแสดงข้อความ "Press left button." หากกดปุ่มขวา mstatus( ) = 2 โปรแกรมจะแสดงข้อความ "Press right button."

4. ในบรรทัดที่ 24 หากเม้าส์เคลื่อนไหวจะตรวจสอบด้วยฟังก์ชัน mmotion( ) ซึ่งจะรีเทอร์นค่าเป็น 0 หากเม้าส์ไม่เคลื่อนที่ และมีค่าเป็น 1 หากเม้าส์มีการเคลื่อนที่ (ดูรูปที่ 6) เมื่อเม้าส์เคลื่อนที่จะตรวจสอบตำแหน่งเม้าส์ในแนวแกน X และ Y ด้วยฟังก์ชัน mgetx( ) และ mgety( ) ในบรรทัดที่ 25 (ดูรูปที่ 7) และบรรทัดที่ 26 และนำไปแสดงผลในบรรทัดที่ 27 เมื่อเราเลื่อนตำแหน่งเม้าส์ตัวเลขแสดงตำแหน่งของเม้าส์จะเปลี่ยนไปเรื่อย ๆ ขอให้ผู้อ่านพิจารณาโปรแกรมและฟังก์ชันพร้อมทั้งดูการทำงานของโปรแกรมประกอบ รูปที่ 8 เป็นผลจากการรันโปรแกรมในรูปที่ 5

การประยุกต์ไปใช้งาน

ในตอนนี้ผู้เขียนขอนำเอาฟังก์ชันและการทำงานของโปรแกรมที่ได้อธิบายมาทั้ง 2 โปรแกรมมารวมกันเพื่อเป็นตัวอย่างให้ท่านผู้อ่านได้นำไปประยุกต์ให้เข้ากับงานของท่าน โดยผู้เขียนจะกำหนดพื้นที่สี่เหลี่ยมขึ้นมา 3 รูป มีสีแดง เขียว และชมพู เมื่อนำเคอร์เซอร์-เม้าส์มากดที่สี่เหลี่ยมสีแดงหรือพื้นที่สีเขียวโปรแกรมจะตรวจเช็คการกดปุ่มเม้าส์ด้านซ้ายว่าตรงกับตำแหน่งพื้นที่ใดและจะบอกว่ากดที่สี่เหลี่ยมสีอะไร หากดเคอร์เซอร์เม้าส์ที่สี่เหลี่ยมสีชมพู (Exit) จะเป็นการออกจากโปรแกรม แต่ถ้าหากกดปุ่มเม้าส์ที่นอกบริเวณพื้นที่ทั้งสามจะมีเสียงสัญญาณสูงต่ำแจ้งให้ทราบ ขอให้ผู้อ่านดูโปรแกรมในรูปที่ 9 ผลจากการรันโปรแกรมในรูปที่ 10 พิจารณาคำอธิบายไปด้วย

คำอธิบายโปรแกรม

1. บรรทัดที่ 7-15 เป็นการวาดพื้นที่สี่เหลี่ยมสีเขียว แดง และชมพูขึ้นมาพร้อมกับคำอธิบายบนพื้นที่สี่เหลี่ยมนั้น บรรทัดที่ 13 กำหนดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์ บรรทัดที่ 14 กำหนดจุดเริ่มต้นของเม้าส์ที่ตำแหน่ง 500,500 บรรทัดที่ 15 กำหนดให้เคอร์เซอร์ของเม้าส์ปรากฏ

2. การเช็คตำแหน่งเม้าส์ใช้ฟังก์ชัน mgetx, mgety ในบรรทัดที่ 19 และ 20 ซึ่งเมื่อมีการกดเม้าส์ที่ตำแหน่งใด ๆ จะนำค่าที่ได้มาเปรียบเทียบในบรรทัดที่ 21, 24 และ 27 ว่าตรงกับข้อแม้ใดก็จะทำตามข้อแม้ที่ถูกต้องนั้น

3. หากกดเม้าส์ไม่ตรงกับบริเวณพื้นที่ทั้งสามนั้นจะได้โทนเสียงต่ำสูงในบรรทัดที่ 31 และ 32

รูปที่ 9
โปรแกรม
CHKBOX.PRG

```
01 : // Program     : chkbox.prg
02 : // Prigrammer : Bhao Singboobpha 7/11/93
03 : // Compile     : CLIPPER chkbox
04 : // Link        : RTLINK FILE chkbox
05 : // Run         : chkbox
06 : sethires(0)
07 : boxfill(300, 300, 150, 100, 0, 4)
08 : boxfill(600, 300, 150, 100, 0, 2)
09 : boxfill(900, 300, 150, 100, 0, 13)
10 : saystring(325, 340, 8, 0, 14, " Red")
11 : saystring(625, 340, 8, 0, 14, "Green")
12 : saystring(930, 340, 8, 0, 14, "Exit")
13 : msetwin(0, 0, 1350, 1000)
14 : mfixpos(500, 500)
15 : mcuron()
16 : DO WHILE .T.
17 :     NButtons := mstatus()
18 :     IF  NButtons = 1
19 :         xpos := mgetx()
20 :         ypos := mgety()
21 :         IF xpos > 300 .AND.  xpos < 450 .AND. ypos > 300 .AND. ypos < 400
22 :               saystring(500, 500, 0, 0, 14, "Press red box   ")
23 :               TONE(330,2)
24 :         ELSEIF xpos > 600 .AND.  xpos < 750 .AND. ypos > 300 .AND. ypos < 400
25 :                saystring(500, 500, 0, 0, 14, "Press green box")
26 :                TONE(660,2)
27 :         ELSEIF xpos > 900 .AND.  xpos < 1050 .AND. ypos > 300 .AND. ypos < 400
28 :                TONE(660,2)
29 :                EXIT
30 :         ELSE
31 :                TONE( 500, 1)
32 :                TONE(1000, 1)
33 :         ENDIF
34 :     ENDIF
35 : ENDDO
36 : mcuroff()
37 : SETTEXT()
38 : RETURN
```

รูปที่ 10 ผลจากการรันโปรแกรมในรูปที่ 9 โดยกดเม้าส์ที่สี่เหลี่ยมแดง

สำหรับตัวอย่างนี้เมื่อผู้อ่านพิจารณาจนเข้าใจแล้วคงสามารถประยุกต์ไปใช้กับงานด้านต่าง ๆ เช่น การทำพูลดาวน์เมนูหรือป็อปอัพเมนู ซึ่งหากนำเอาการแสดงผลภาษาไทย 3 ระดับในกราฟิกและการรับภาษาไทยในกราฟิกโหมดที่ผู้เขียนเสนอมาในฉบับก่อน คงจะช่วยให้ผู้เขียนโปรแกรมคลิปเปอร์ทำงานทางด้านการนำเสนอ แบบเรียนคอมพิวเตอร์ หรือฐานข้อมูลทางด้านภาพที่เป็นกราฟิกได้ ต่อไปผู้เขียนขอเสนอการเขียนโปรแกรมแสดงปุ่มกดที่เรียกว่าบัททอนหรือแป้นสามมิติและการนำไปประยุกต์ใช้งานกับฐานข้อมูลรูปภาพ

สร้างฟังก์ชันสำหรับแป้นสามมิติ

เพื่อให้การเขียนโปรแกรมและการเรียกใช้ง่ายขึ้นผู้เขียนจึงได้สร้างฟังก์ชันสำหรับแสดงแป้นสามมิติที่แสดงอักษรบนแป้น มีสองแบบคือแบบนูนเมื่อยังไม่ถูกกดและแบบแบนเมื่อถูกกดแล้ว ซึ่งตรงนี้เป็นส่วนที่เราจะหลอกให้คนดูจอภาพเห็นคล้อยตามเรามีแป้นที่นูนออกมาและแบนลงไปเมื่อถูกกดดังนี้

1. แป้นจะมีลักษณะเป็นสี่เหลี่ยม โดยส่วนนี้ผู้ที่เรียกใช้ฟังก์ชันสามารถกำหนดสีพื้นได้

2. ขอบของพื้นด้านบนและด้านซ้ายมือจะถูกกำหนดด้วยสีขาวจ้า ส่วนขอบทางด้านขวาและทางด้านล่างถูกกำหนดด้วยสีขาว

3. ส่วนการกำหนดให้แป้นแลดูเหมือนถูกกดลงไปนั้นให้กำหนดขอบทั้งสี่เป็นสีดำหรือสีเข้มเท่านั้น หากท่านผู้อ่านไม่พอใจก็สามารถปรับสีในฟังก์ชันนี้ได้ ในการวาดเส้นเพื่อให้เกิดแป้นสามมิตินี้ผู้เขียนใช้วิธีการตีเส้นอย่างง่าย ๆ เพื่อให้โปรแกรมทำงานได้ไวขึ้น

รูปที่ 11
ฟังก์ชัน normbutt( )
และ presbutt( )

```
059 : FUNCTION normbutt(x, y, width, high, char, cBotton)
060 : boxfill( x, y, width, high, 0, 0)
061 : clrwin( x, y, x+width, y+high)
062 : boxfill( x+4, y+4, width-8, high-8, 0, cBotton)
063 : drawline(x+5, y+high-5, x+width-4, y+high-5, 0, 0, 15)
064 : drawline(x+6, y+high-6, x+width-6, y+high-6, 0, 0, 15)
065 : drawline(x+8, y+high-8, x+width-8, y+high-8, 0, 0, 15)
066 : drawline(x+10, y+high-10, x+width-10, y+high-10, 0, 0, 15)
067 : drawline(x+5, y+6, x+5, y+high-10, 0, 0, 15)
068 : drawline(x+7, y+8, x+7, y+high-12, 0, 0, 15)
069 : drawline(x+9, y+10, x+9, y+high-14, 0, 0, 15)
070 : drawline(x+width-4, y+5, x+width-4, y+high-5, 0, 0, 7)
071 : drawline(x+width-6, y+8, x+width-6, y+high-8, 0, 0, 7)
072 : drawline(x+width-8, y+11, x+width-8, y+high-11, 0, 0, 7)
073 : drawline(x+6, y+4, x+width-4, y+4, 0, 0, 7)
```

→

รูปที่ 11 (ต่อ)

```
074 : drawline(x+8, y+6, x+width-6, y+6, 0, 0, 7)
075 : drawline(x+10, y+8, x+width-8, y+8 , 0, 0, 7)
076 : saystring(x+(width/2)-((LEN(char)/2)*18), y+(high/2)-18, 32, 0, 0, char)
077 : RETURN NIL
079 : //*****************************************************************
080 : FUNCTION presbutt(x, y, width, high, char, cBotton)
081 : boxfill( x, y, width, high, 0, 0)
082 : boxfill( x, y, width, high, 0, cBotton)
083 : drawline(x, y, x+width, y, 3, 0, 0)
084 : drawline(x, y+1, x+width, y+1, 3, 0, 0)
085 : drawline(x, y+2, x+width, y+2, 3, 0, 0)
086 : drawline(x, y+3, x+width, y+3, 3, 0, 0)
087 : drawline(x, y+high, x+width, y+high, 3, 0, 0)
088 : drawline(x, y+high-1, x+width, y+high-1, 3, 0, 0)
089 : drawline(x, y+high-2, x+width, y+high-2, 3, 0, 0)
090 : drawline(x, y+high-3, x+width, y+high-3, 3, 0, 0)
091 : drawline(x, y, x, y+high, 3, 0, 0)
092 : drawline(x+1, y, x+1, y+high, 3, 0, 0)
093 : drawline(x+2, y, x+2, y+high, 3, 0, 0)
094 : drawline(x+width, y, x+width, y+high, 3, 0, 0)
095 : drawline(x+width-1, y, x+width-1, y+high, 3, 0, 0)
096 : drawline(x+width-2, y, x+width-2, y+high, 3, 0, 0)
097 : saystring((x+(width/2)+2)-((LEN(char)/2)*18),y+(high/2)-20,32,0,0,char)
098 : RETURN NIL
```

สำหรับฟังก์ชันและการเรียกใช้ให้ดูรูปที่ 11

ในบรรทัดที่ 76 และบรรทัดที่ 97 เป็นการกำหนดให้อักษรที่จะแสดงบนแป้นสามมิติอยู่ตรงกลางแป้น สำหรับฟังก์ชัน normbutt( ) นั้นหากกำหนดให้มีขนาดใหญ่ก็จะเป็นกระดานสามมิติได้ หากผู้อ่านต้องการที่จะแสดงข้อความภาษาไทยบนแป้นสามมิตินี้ให้เรียกใช้ฟังก์ชัน saythai( ) แทนฟังก์ชัน saystring( ) (ดูรูปที่ 12) รูปที่ 13 แสดงแป้นสามมิติทั้งสองอย่าง

```
normbutt(x, y, width, high, char, cBotton)
presbutt(x, y, width, high, char, cBotton)
    X           X of bottom left
    Y           Y of bottom left
    width       width
    high        height
    char        string on bottom
    cBotton     color of button 0-15
```

รูปที่ 12 ฟังก์ชัน normbutt( ) หากมีขนาดใหญ่ก็จะเป็นกระดานสามมิติได้

รูปที่ 13 แสดงแป้นสามมิติแบบนูนและแบบถูกกด

ตัวอย่างการนำไปใช้

ผู้เขียนได้ให้ตัวอย่างการนำไปใช้งานโดยการนำรูปภาพพร้อมกับฐานข้อมูลของรูปภาพมาแสดงประกอบกัน โปรแกรมในรูปที่ 14 เมื่อรันโปรแกรมจะได้ผลในรูปที่ 15 เมื่อนำเม้าส์ไปคลิ๊กที่ปุ่มแสดงชื่อของเด็กจะมีรูปเด็กปรากฏขึ้นดังรูปที่ 16 ในตอนที่คลิ๊กเม้าส์ปุ่มที่ถูกกดจะมีสภาพเหมือนยุบลงไป เมื่อกดที่ปุ่ม OK ภาพเด็กจะหายไป เราสามารถเลือกฐานข้อมูลรูปภาพเหล่านี้ได้จากการใช้เม้าส์ สำหรับฐานข้อมูลและรูปเด็กเหล่านี้ได้จากโปรแกรม Q+E Database/VB ซึ่งเคยแนะนำในหนังสือ

รูปที่ 14 โปรแกรม CHECKPIC.PRG แสดงรูปพร้อมประวัติเด็ก

```
001 : // Program     : checkpic.prg
002 : // Prigrammer : Bhao Singboobpha 7/11/93
003 : // Compile     : CLIPPER checkpic
004 : // Link        : RTLINK FILE checkpic
005 : // Run         : checkpic
006 : sethires(0)
007 : //boxfill( 0, 0, 1350, 1000, 0, 2)
008 : //boxfill( 10,  10, 1330, 980, 0, 0)
009 : msetwin( 10, 10, 1330, 980)
010 : mfixpos(675, 500)
011 : mcuron()
012 : normbutt( 10,  830, 1330, 160, "Sample For Mouse & Button Control", 4)
013 : normbutt( 10,  730,  220, 100, "Heather", 2)
014 : normbutt(231,  730,  220, 100, "David", 2)
015 : normbutt(451,  730,  220, 100, "Lauren", 2)
016 : normbutt(671,  730,  220, 100, "Aaron", 2)
017 : normbutt(891,  730,  220, 100, "Calvin", 2)
018 : normbutt(1111, 730,  229, 100, "John", 2)
019 : normbutt(  10,  10, 1330, 100, "Quit", 4)
020 : DO WHILE .T.
021 :   NButtons := mstatus()
```

→

รูปที่ 14 (ต่อ)

```
022 :   IF  NButtons = 1
023 :       xpos := mgetx()
024 :       ypos := mgety()
025 :       IF     xpos >=   11 .AND. xpos <=  230 .AND. ypos >= 730 .AND. ypos <=830
026 :          presbutt( 10, 730, 220, 100, "Heather", 2)
027 :          showpic(1)
028 :          normbutt( 10, 730, 220, 100, "Heather", 2)
029 :       ELSEIF xpos >=  231 .AND. xpos <=  450 .AND. ypos >= 730 .AND. ypos <=830
030 :          presbutt(231, 730, 220, 100, "David", 2)
031 :          showpic(2)
032 :          normbutt(231, 730, 220, 100, "David", 2)
033 :       ELSEIF xpos >=  451 .AND. xpos <=  670 .AND. ypos >= 730 .AND. ypos <=830
034 :          presbutt(451, 730, 220, 100, "Lauren", 2)
035 :          showpic(3)
036 :          normbutt(451, 730, 220, 100, "Lauren", 2)
037 :       ELSEIF xpos >=  671 .AND. xpos <=  890 .AND. ypos >= 730 .AND. ypos <=830
038 :          presbutt(671, 730, 220, 100, "Aaron", 2)
039 :          showpic(4)
040 :          normbutt(671, 730, 220, 100, "Aaron", 2)
041 :       ELSEIF xpos >=  891 .AND. xpos <= 1100 .AND. ypos >= 730 .AND. ypos <=830
042 :          presbutt(891, 730, 220, 100, "Calvin", 2)
043 :          showpic(5)
044 :          normbutt(891, 730, 220, 100, "Calvin", 2)
045 :       ELSEIF xpos >= 1101 .AND. xpos <= 1330 .AND. ypos >= 730 .AND. ypos <=830
046 :          presbutt(1111, 730, 220, 100, "John", 2)
047 :          showpic(6)
048 :          normbutt(1111, 730, 220, 100, "John", 2)
049 :       ELSEIF xpos >=10 .AND. xpos <=1330 .AND. ypos >=10 .AND. ypos <= 110
050 :          EXIT
051 :       ENDIF
052 :   ENDIF
053 : ENDDO
054 : mcuroff()
055 : SETTEXT()
056 : RETURN
057 :
058 : //*****************************************************************
059 : FUNCTION normbutt(x, y, width, high, char, cBotton)
    *********
```

→

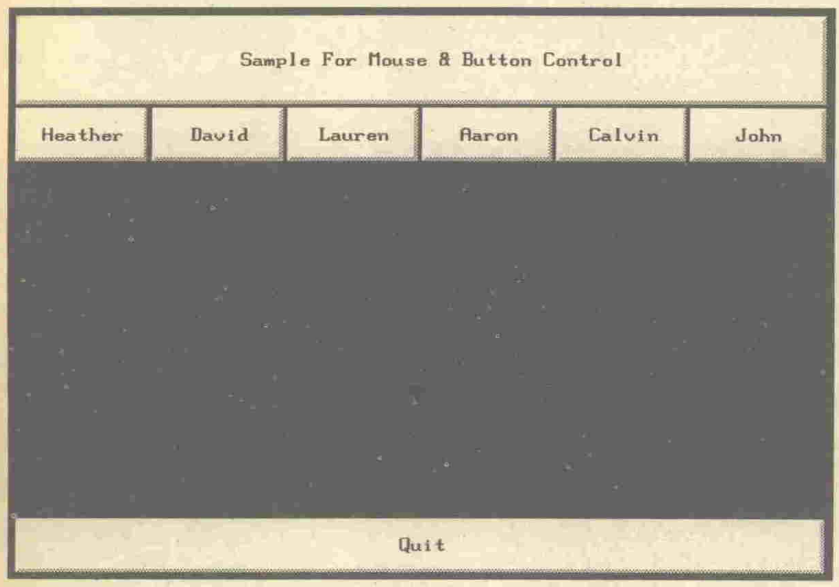
รูปที่ 14 (ต่อ)

```
077 : RETURN NIL
079 : //*****************************************************************
080 : FUNCTION presbutt(x, y, width, high, char, cBotton)
    ***********
098 : RETURN NIL
099 :
100 : //*****************************************************************
101 : FUNCTION showpic(record)
102 : LOCAL cButtons
103 : DO CASE
104 :    CASE record = 1
105 :         picread(200, 200, 0, "heather.pcx")
106 :    CASE record = 2
107 :         picread(200, 200, 0, "david.pcx")
108 :    CASE record = 3
109 :         picread(200, 200, 0, "lauren.pcx")
110 :    CASE record = 4
111 :         picread(200, 200, 0, "aaron.pcx")
112 :    CASE record = 5
113 :         picread(200, 200, 0, "calvin.pcx")
114 :    CASE record = 6
115 :         picread(200, 200, 0, "john.pcx")
116 : ENDCASE
117 : USE Daycare.dbf
118 : GO record
119 : saystring(600, 630, 4, 0, 14, "FIRSTNAME : " + first_name)
120 : saystring(600, 555, 4, 0, 14, "LASTNAME  : " + last_name)
121 : saystring(600, 480, 4, 0, 14, "PARENT    : " + parent)
122 : saystring(600, 405, 4, 0, 14, "OCCUPATION: " + occupation)
123 : saystring(600, 330, 4, 0, 14, "BIRTHDAY  : " + DTOC(birth_date))
124 : USE
125 : normbutt(650, 220, 250, 80, "OK.", 4)
126 : DO WHILE .T.
127 :    cButtons := mstatus()
128 :    IF  cButtons = 1
129 :        xpos := mgetx()
130 :        ypos := mgety()
131 :        IF xpos >= 650 .AND. xpos <=  900 .AND. ypos >= 220 .AND. ypos <=300
132 :           EXIT
```

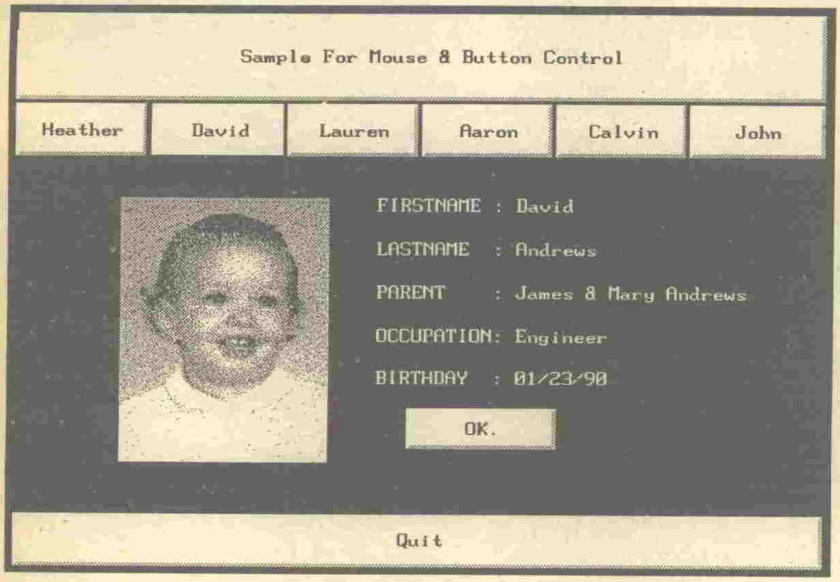
→

รูปที่ 14 (ต่อ)

```
133 :        ENDIF
134 :    ENDIF
135 : ENDDO
136 : clrwin(20, 200, 1300, 720)
137 : //boxfill( 20,  200, 1310, 520, 0, 1)
138 : RETURN NIL
```



รูปที่ 15 ผลจากการรันโปรแกรมที่ 14



รูปที่ 16 ผลจากการกดปุ่มเม้าส์ที่รายชื่อเด็ก

ไมโครคอมพิวเตอร์เล่มที่ 103 แต่อยู่ในฟอร์แมต BMP ผู้เขียนได้แปลงให้อยู่ในฟอร์แมต PCX 16 สี เพื่อให้สามารถนำมาแสดงได้ (มีอยู่ในซอฟต์เมล์) ให้ผู้อ่านพิจารณาไล่โปรแกรมทำความเข้าใจ เพื่อให้สามารถนำไปประยุกต์ในเรื่องของฐานข้อมูลที่เกี่ยวข้องกับภาพกราฟิก

ของแถม

สำหรับซอฟต์เมล์ที่ผู้เขียนเตรียมให้ในฉบับนี้จัดอยู่ในแฟ้ม MOUSE.EXE ซึ่งเป็นแฟ้มที่บีบอัด ตัวอย่างโปรแกรม รูปภาพเด็กในฟอร์แมต PCX แล้วยังมีไลบรารีที่ผู้เขียนได้ปรับปรุงให้เรียกใช้ฟังก์ชันของ DGE ได้ทันที และนอกจากนี้ผู้เขียนยังได้ให้แฟ้มข้อมูลรายละเอียดของฟังก์ชัน DGE (เมื่อระเบิดแฟ้มออกมาจะได้แฟ้มที่มีชื่อ DGE.1-DGE.4) ทั้งหมดซึ่งสามารถสั่งพิมพ์ออกมาโดยราชวิถีเวิร์ด (ผู้เขียนใช้ .PA แบ่งหน้ากระดาษ) ในฉบับหน้าผู้เขียนมีเรื่องที่น่าสนใจของคลิปเปอร์ที่สนุก ๆ มาฝากอีกเช่นเคยคอยติดตามนะครับ